

Article

Downlink Performance Modeling and Evaluation of Batteryless Low Power BLE Node

Ashish Kumar Sultania ¹, Carmen Delgado ², Chris Blondia ¹ and Jeroen Famaey ^{1,*}

¹ IDLab-Department of Computer Science, University of Antwerp-imec, 2000 Antwerp, Belgium; ashishkumar.sultania@uantwerpen.be (A.K.S.); chris.blondia@uantwerpen.be (C.B.)

² i2CAT Foundation, 08034 Barcelona, Spain; carmen.delgado@i2cat.net

* Correspondence: jeroen.famaey@uantwerpen.be; Tel.: +32-4726-449-37

Abstract: Deploying low maintenance and long-life systems is an important requirement of emerging commercial Internet of Things (IoT) solutions. Such systems can be envisioned in which the connected devices are powered by energy harvested from ambient sources and stored in long-lifetime capacitors rather than short-lived and polluting batteries. However, due to the unpredictable nature of ambient energy harvesting, such batteryless IoT devices might not always have enough energy to initiate communication. The Bluetooth Low Energy (BLE) specification defines support for Low Power Nodes (LPNs) using the friendship feature, where the LPN is associated with a neighbouring friend node (FN). The LPN can receive downlink (DL) data and remain connected to the network via the FN that buffers the LPN's incoming packets while allowing the LPN to save energy by sleeping or turning itself off. This novel BLE feature makes the LPN highly suitable to support the connection of batteryless ambiently-powered IoT devices. While the LPN can decide when to transmit uplink (UL) packets and does not depend on the FN to receive downlink (DL) data, the LPN needs to poll its FN to receive the buffered packets. However, the DL packet latency increases with this process due to the buffering time at the FN. Therefore, in this work, we present an analytical model to characterize the performance as a function of DL data latency and packet delivery ratio (PDR) of a batteryless LPN powered by different harvesting powers and capacitor sizes. This would help to optimally choose the correct configuration of the batteryless LPN for its network deployment. We also compare the analytical model and simulation results, showing consistency with an average error of 2.23% for DL data latency and 0.09% for the PDR.

Keywords: batteryless; Bluetooth Low Energy; energy harvesting; IoT; low power node



Citation: Sultania, A.K.; Delgado, C.; Blondia C.; Famaey, J. Downlink Performance Modeling and Evaluation of Batteryless Low Power BLE Node. *Sensors* **2022**, *22*, 2841. <https://doi.org/10.3390/s22082841>

Academic Editors: Carlo Trigona, Olfa Kanoun and Slim Naifar

Received: 31 January 2022

Accepted: 5 April 2022

Published: 7 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The emerging Internet of Things (IoT) concept aims to wirelessly connect billions of devices to the internet. Usually, these devices are powered by batteries. However, replacing batteries for such a large number of devices increases the cost of system maintenance, as it requires manual intervention. Battery life is also reduced due to over-charging and over-discharging, as well as environmental conditions (e.g., temperature fluctuations) [1]. Moreover, disposing of batteries is also a serious environmental concern. Therefore, considering the downside of battery usage and the requirement of low maintenance 'deploy and forget' devices for the IoT, batteries can be replaced by easily recyclable and cheap capacitors combined with energy harvesting to obtain and store energy in a more sustainable way [2,3]. Such devices can achieve a long-term multi-decade active lifetime.

Bluetooth Low Energy (BLE) has become one of the most important technologies for the IoT thanks to its features (i.e., extremely low power consumption, low cost, and mesh topology support). It supports many use-cases, such as asset tracking, sensing, monitoring, control, and automation, with thousands of interconnected devices [4]. However, most existing BLE-based IoT solutions consider battery-powered devices. Meanwhile, designing a batteryless solution is important for long-term sustainability (i.e., no adverse impact on

the environment), but it is also a challenging task. Batteryless devices usually charge their capacitor by harvesting ambient energy, which is often uncontrollable and unpredictable, allowing them to only work for a short amount of time. These conditions can result in intermittent behaviour where the device can experience multiple power failures. The duration of the power failure cycle depends on the harvesting power, capacitor size, and operating voltage. Moreover, for a specific harvesting power, the time to charge a capacitor increases with its capacitance [5]. Therefore, choosing the optimal capacitor size based on the minimum available harvesting power and required operations to achieve the desired performance is important.

The BLE specification [6] already provides a friendship feature to aid Low Power Nodes (LPNs) to save energy, keeping themselves in SLEEP or OFF mode for most of the time. The LPN awakes only (or turns on) to transmit or receive data packets. For sending uplink (UL) data packets, the LPN can broadcast them at any time. On the other hand, the LPN can start a poll process (request/response) to receive the downlink (DL) packets that are temporarily buffered at the friend node (FN). The polling happens periodically to receive any incoming data packets. This can reduce the overall energy consumption at the cost of increased DL data latency.

In battery-powered nodes (non-harvesting), long-term energy consumption is a key factor as it affects the total lifetime of the device. This factor is less important in batteryless devices powered by ambient energy as they can perpetually recharge themselves. Instead, short-term energy consumption bursts are more important as they may drain the capacitor and result in a power failure. In this article, we focus on analyzing the achievable DL latency and packet delivery ratio (PDR) using a batteryless BLE LPN supported by different harvesting powers and capacitor sizes. We model the system utilizing a discrete-time Markov Chain and compare the results with our developed simulator [7]. The simulator needs to simulate extended periods of time and be repeated many times to average out stochastic variability, whereas a model can immediately provide the results for a specific use case at steady state. This allows the model, compared to a full-fledged simulator, to be integrated into a real-time optimizer in the node that can help to easily reconfigure the BLE configuration parameters based on current conditions. The model can be used to theoretically perform a feasibility analysis of the friendship feature for bidirectional batteryless communication and to deduce its limitations. Moreover, it would aid in designing battery-less BLE devices by calculating its optimal hardware and software configuration for IoT use-cases that require downlink or bidirectional communication. Specifically, our model can be used to optimally derive the hardware configuration in terms of capacitor size. The software configuration parameters, in terms of ReceiveDelay, ReceiveWindow, Tx power level, and Poll interval of the batteryless LPN can be determined based on the use-case requirement in terms of data rate, DL latency, and PDR considering the neighboring FN queue size and minimum harvesting power at the network site.

The paper is organized as follows: Section 2 summarizes related work; Section 3 gives an overview of the BLE friendship feature; Section 4 explains the batteryless LPN system; and Section 5 presents the analytical model. Subsequently, Section 6 presents all the evaluation results and Section 7 shows the usage of model to derive optimal parameters by considering a logistic management IoT use-case. Section 8 draws the main conclusions. Finally, Section 9 briefly discusses limitations of the model and potential avenues for future work.

2. Related Work

There are several batteryless BLE prototypes designed by researchers. Silva et al. [8] presented a prototype where an implantable batteryless glucose monitor node was capable of transmitting data to a mobile phone when supplied power wirelessly using a Radio Frequency (RF) energy source. However, their design did not consider the Bluetooth mesh friendship feature and sent only UL data. Liu et al. recently proposed another RF-power based prototype for indoor BLE beaconing [9]. They tested the device for both UL and DL

data. However, they did not analyze the effect of the capacitor size nor the performance in terms of DL latency. Fraternali et al. [10] designed a batteryless BLE sensor node based on ambient light energy harvesting. They presented a predictive algorithm that adapted to the changing ambient conditions to maximize the node lifetime and application quality of service. However, their focus was also only on the UL data. Nilsson et al. also conducted indoor and outdoor experiments to evaluate power consumption, range, throughput, and UL latency of a Bluetooth mesh network [11]. In summary, most of the research on batteryless BLE nodes is performed considering UL data without considering the friendship feature or DL communications.

There are two papers that considered the friendship feature. However, they considered battery-powered LPNs rather than batteryless LPNs. The first paper is by Alvarez et al. [12], who performed an in-depth analysis of security on the friendship concept. They demonstrated that a denial-of-friendship attack could reduce the LPN's battery life by 70-fold. The second paper is by Darroudi et al. [13]. They presented an analytical model to study the energy performance parameters of a battery-operated BLE LPN. These parameters include current consumption, lifetime, and energy consumed per delivered bit for different PollTimeout and ReceiveWindow values. They calculated the evaluation parameters based on the average value of time and current consumption for all the radio states and for fixed UL data rate interval, in contrast to DL traffic, which does not require the polling feature or friend queue.

In contrast, our work is focused on the DL latency and packet delivery metrics. We focus on a batteryless LPN harvesting energy from ambient sources and not on battery power LPNs. As such, we do not consider current consumption as the evaluation metric, but rather a constraint that should be satisfied. Moreover, we focus on DL packet arrivals, which are first queued at the FN, and so our model studies the queueing performance based on the Markov chain with Poisson distributed data arrivals which provides discrete random distribution of data. In real-world applications, the data arrivals are also random, and so it is used to represent the message arrivals in the BLE network. The model provides the opportunity to analyze the batteryless LPN powered by different capacitor sizes receiving different harvested powers for various Bluetooth mesh network parameters, such as friend queue size, packet size, or receive delay. The model is also compared to our developed simulator as presented in [7].

3. BLE Friendship Feature

The Bluetooth mesh network is based on a publish-subscribe model for data communication. It supports up to 32,767 devices with a maximum of 128 hops [14]. The devices need to have addresses and security information to become part of the mesh network to send or receive data. This is enabled by a provisioner, and the provisioned device is called anode. Any node can support four additional optional features (namely, relay, proxy, friend, and low power [6]), as shown in Figure 1.

The relay node supports receiving and re-transmitting data packets to all the nodes in its connection range. It helps to extend the range of the entire network. The proxy node helps the non-mesh-supported BLE devices to communicate via the mesh network. To enable both these features, the node needs to listen continuously for advertising packets and, therefore, should be connected to a continuous power source [15]. To support energy-constraint nodes, BLE introduces the LPN, which can offload the task of listening to its incoming channel to a FN that can temporarily buffer any incoming data packets. For example, in Figure 1, LPN 'A' is associated with FN 'X' for the DL data. However, it does not need help to transmit UL data. The LPN can wake up any time to send UL data, assuming that at least one of its neighbouring nodes actively listens to the incoming channels. With the help of its FN, the LPN can save much of its energy by turning off its components and keeping itself in SLEEP mode for most of the time. However, before switching to the SLEEP or OFF state, the LPN needs to first establish a friendship with one of its single-hop neighbouring nodes. This neighbour needs to support the optional friendship feature.

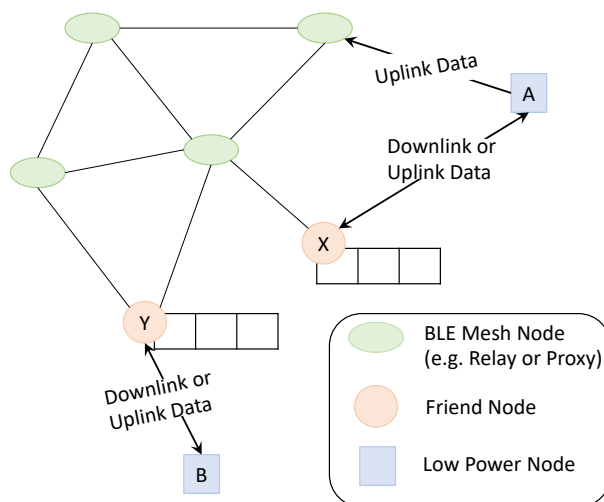


Figure 1. BLE mesh network.

3.1. Friendship Establishment

Figure 2 illustrates the messages exchanged during the friendship establishment procedure. The same Friend Request is received by one or more neighbouring friend feature enabled nodes. This message is 28 bytes of a network Protocol Data Unit (PDU), which consists of Criteria, ReceiveDelay (RD), PollTimeout (PT), PreviousAddress, NumElements, and NumReqCounter [6].

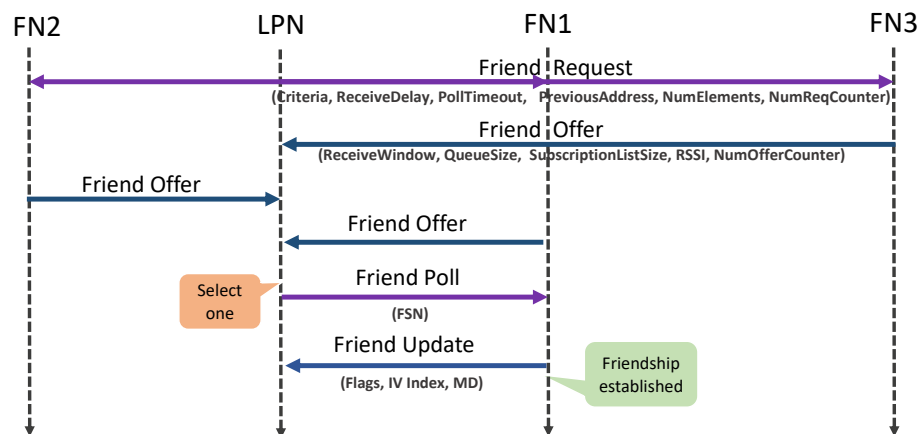


Figure 2. Friendship establishment procedure of BLE.

The Criteria is the minimum requirement in terms of the Received Signal Strength Indicator (RSSI), ReceiveWindow (RW), and supported queue size that an FN should satisfy. The RD is the waiting time of the LPN before it should expect to receive a poll response from the FN. PT is used to ensure that the FN does not keep the friendship context indefinitely and terminates the relationship if it does not receive any poll before the PT timer expires (since the last poll). Lastly, the RW is the maximum time that the LPN should listen to receive a response from the FN after sending a poll request. These timers can be seen in Figure 3. The RD and RW can be configured with a maximum value of 255 ms and a PT with a maximum of 96 h. PreviousAddress defines the address of its previous FN, NumElements defines its number of addressable elements (as messages in the BLE mesh network are exchanged from the elements), and NumReqCounter defines the number of sent requests required to calculate the Friendship Security Material.

The FNs that satisfy the initial criteria defined by the LPN will offer the friendship by sending a Friend Offer (FO). This response message has a network PDU of 24 bytes, which consists of RW, supported friend queue (FQ) size, size of subscription list, calculated

RSSI, and number of FO sent by this FN. Based on all the received FOs, the LPN selects the best among them, considering the Criteria satisfactions. To confirm this selection, the LPN acknowledges the selected FN with a Friend Poll (FP) message of 19 bytes, with the Friend Sequence Number (FSN) equal to zero. Next, the FN finalizes the friendship procedure by sending a Friend Update (FU) message of 24 bytes that contains the flags, More Data (MD), indicating if the queue is empty or not, and another flag that indicates the update of the initialization vector (IV) index (which is a shared network resource) or security key. It also contains the current IV index value known by the FN. Once the friendship is established, the LPN also notifies the FN with all the subscribed group list addresses so that the FN knows which messages it should buffer and forward to the LPN.

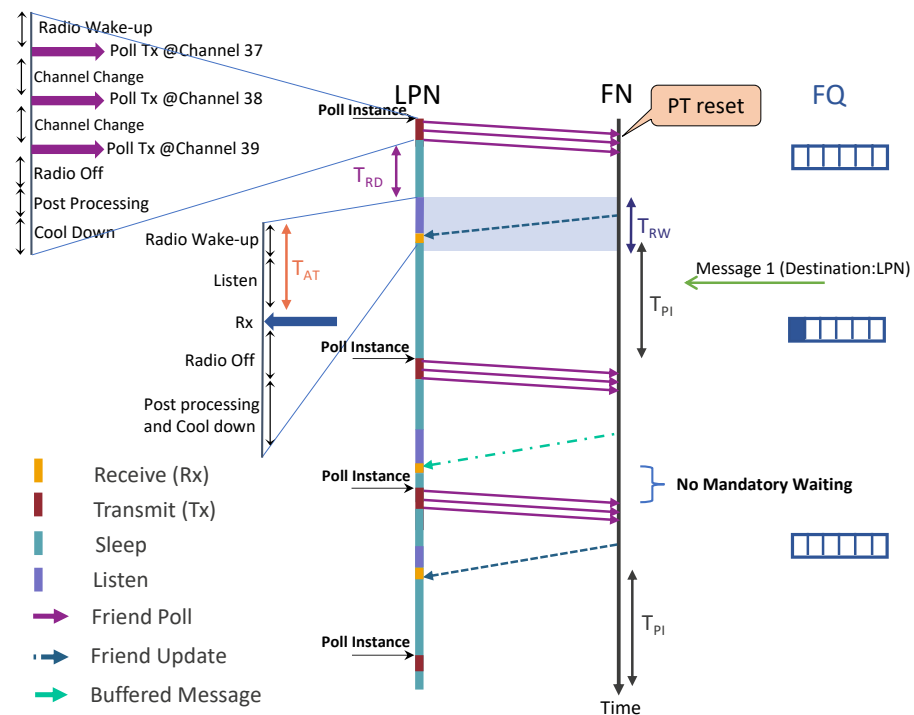


Figure 3. Sequence diagram of the messages exchanged between the LPN and FN to receive DL packets.

3.2. Receiving Downlink Data

As the friendship is established, the LPN can now switch to SLEEP mode, and all the incoming DL packets of the LPN will be buffered in the FQ. The process is shown in Figure 3.

The LPN can retrieve them from the FN by periodically waking up and sending an FP message in all three broadcast channels (37 to 39). On reception of the FP, the FN gets a time window of RW (T_{RW}) seconds, starting (T_{RD}) seconds after receiving the FP to send a response [16]. We also define an arrival time (AT or T_{AT}), which denotes the total listening time until the LPN starts receiving the message. While sending the FP, the LPN needs to toggle the FSN value to indicate the acknowledgement of the previous message. If the FN receives the same FSN field value as the last FP message received from the FN, it responds with the same message that it has previously sent, unless the FQ head is updated. This can be updated by any new incoming packet(s) whenever it arrives at a full FQ. In this case, the oldest packet (FQ head) would be discarded (dequeued) and the newly arrived packet would be added in the queue tail (queued). This strategy is called ‘push-out’. Therefore, if the previously sent message has been discarded (on the arrival of new packets), then the oldest entry in the FQ shall be sent. If the FQ is empty, an FU is sent with MD equal to zero. To avoid receiving frequent, continuous FU messages, a timer poll interval timer (T_{PI}) can be considered. On receiving an FU message, an LPN needs to wait until the PI expires to send the next FP message. As we consider a batteryless LPN, it may happen

that it is unable to send the FP exactly at the expiration of PI due to energy constraints. The friendship can also be terminated if the LPN does not receive any response from the FN after sending the FP for a certain amount of attempts. Moreover, BLE mesh provides infrastructure-less connectivity without any overheads of time synchronization [17]. As the nodes (except LPNs) are assumed to be always listening, there is no need to synchronize their listening timeslots. Moreover, the LPN always initiates communication with the FN itself (i.e., receiver-initiated communication); therefore, the LPN does not need to synchronize explicitly for receiving data either [16]. Timing of the RW is done relative to the FP message sent by the LPN to its FN.

4. Batteryless BLE LPN

Let the LPN have an already established friendship with a neighbouring FN. To receive DL packets, the radio of an LPN can be in 15 different states, as shown in Figure 3 (radio wake-up to post-processing and cool down). The voltage (energy) of a capacitor as a function of time, supporting various states of the radio as used in the remainder of this section, was first introduced by Delgado et al. [18].

4.1. Capacitor Voltage

Equation (1) provides the voltage on the capacitor after the device operates at a load current of $I_L(X)$ (amperes) in state X for a time Δt , harvesting from a constant voltage source, providing a power equal to P_h (watts).

$$V(t + \Delta t) = \frac{V_{max} \cdot P_h}{P_h + V_{max} I_L(X)} \left(1 - e^{-\frac{\Delta t \cdot (P_h + V_{max} I_L(X))}{V_{max}^2 \cdot C}} \right) + V(t) \cdot e^{-\frac{\Delta t \cdot (P_h + V_{max} I_L(X))}{V_{max}^2 \cdot C}}, \quad (1)$$

where, V_{max} (volt) is the maximum capacitor voltage for capacitance C (farad).

As the capacitor has a small energy density and due to the unpredictable availability of harvested energy, it can result in intermittent on-off behaviour of the batteryless LPN, as shown in Figure 4. When the capacitor voltage drops below the minimum operating voltage (V_{cutoff}), the LPN transitions to the OFF state. If the LPN is in the OFF state, it transitions to the SLEEP state only when achieving a predefined turn-on voltage V_{turnon} .

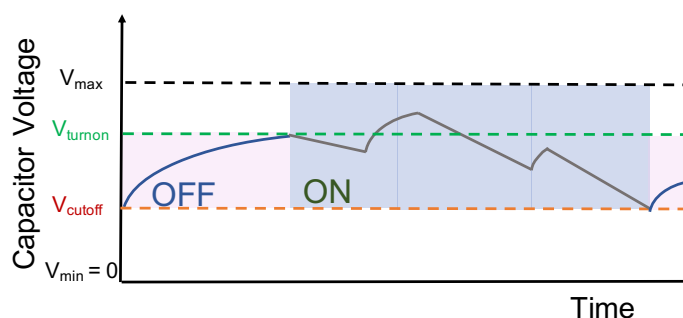


Figure 4. Intermittent behaviour of a batteryless LPN harvesting continuously.

However, in our model, an energy-aware LPN is considered that tries to avoid the OFF state. It can use an ultra-low-power comparator, with power consumption in the order of pico-watts [19], which can determine if a defined voltage level in the capacitor has been reached. This voltage could be set to the minimum voltage level required to execute a task (such as poll, receive data) without going below V_{cutoff} . We refer to that approach as an energy-aware device.

As stated before, the friendship can be terminated if no response is received either after sending N FPs or if the time between two consecutive FPs is larger than T_{PT} . To avoid this, the P_h is considered such that it allows the capacitor to recharge itself from V_{min} to a threshold voltage level that is sufficient to perform a poll request-response cycle in less than T_{PT} . The time to send the FP when the batteryless LPN needs to have a required

threshold voltage of V_{th}^{FP} to start the FP and the remaining voltage of the capacitor after an FP is calculated in Section 4.2.

4.2. Time to Next Poll

The time at which the next FP can take place depends on whether the FQ is empty or not and the available energy after the previous poll.

4.2.1. FQ Is Not Empty

The voltage of the capacitor after all the polling stages can be calculated according to Equation (1), starting from an initial voltage (V_0) and considering 'radio wake-up' as an initial state. Let V_f be the final voltage after all the polling radio stages (as shown in Figure 3) of the previous poll are executed. Therefore, there are two scenarios:

- $V_f \geq V_{th}^{FP}$: The next FP can be sent immediately, and the time between two FPs (T_{FP}) is the sum of all the state timings as defined in Table 1 and is calculated as Equation (2). The capacitor voltage at the start of the next poll instance will be V_f .

$$T_{FP} = \sum_{i=AllStates} T_i \quad (2)$$

- $V_f < V_{th}^{FP}$: As V_f is less than the required voltage to poll, the device needs to switch to SLEEP mode to charge the capacitor and avoid its voltage from reaching V_{cutoff} . Let the node sleep for the time T_{sleep} which can be deduced using Equation (1). Therefore, the time between two FPs has an extra wait time of T_{sleep} in addition to all the FP state's timings and is calculated in Equation (3).

$$T_{FP} = \sum_{i=AllStates} T_i + T_{sleep} \quad (3)$$

The capacitor voltage at the start of the next poll instance will be V_{th}^{FP} (after the capacitor is charged while the LPN is in SLEEP mode).

4.2.2. FQ Is Empty

If the FQ is empty, the LPN needs to wait for the time PI (T_{PI}) to poll, as it received an FU after the previous FP message. Therefore, in this case, the voltage V_f would be calculated after a sleep time of T_{PI} . Now again, there are two cases.

- $V_f \geq V_{th}^{FP}$: The time to poll has an extra delay of T_{PI} in addition to all the FP state's timings.

$$T_{FP} = \sum_{i=AllStates} T_i + T_{PI} \quad (4)$$

The capacitor voltage at the start of the next poll instance will be V_f .

- $V_f < V_{th}^{FP}$: Similar to above, the node needs to switch to SLEEP mode to provide enough time to charge the capacitor. Therefore, the time between two FPs has an additional time T_{sleep} and T_{PI} in addition to all the FP state's timings.

$$T_{FP} = \sum_{i=AllStates} T_i + T_{PI} + T_{sleep} \quad (5)$$

The capacitor voltage at the start of the next poll instance is V_{th}^{FP} .

The threshold voltage of the FP (V_{th}^{FP}) can be calculated by taking the inverse of the function, as mentioned in Equation (1) calculating $V(t)$, knowing the value of $V(t + \Delta t)$ for all the states of the poll, considering the initial voltage as V_{cutoff} , and with starting state 'post-processing and cool down'.

Table 1. State sequence for a poll request and response of a LPN.

Identifier	Radio State	Time [μ s] of the LPN [mA]	Current Cons.	Comments
1	Radio wake-up	2940	0.587	Prepare for poll transmission
2	TX (at 4 dBm)	471	9.09	Tx advertisement packet on channel 37
3	Change channel	29	7.78	Change radio frequency
4	TX (at 4 dBm)	380	9.15	Tx advertisement packet on channel 38
5	Change channel	29	7.57	Change again the radio frequency
6	TX (at 4 dBm)	384	9.16	Tx advertisement packet on channel 39
7	Radio off	34	6.60	Main radio in OFF state
8	Post processing	420	2.17	
9	Cool down	20,820	0.00653	Prepare to switch to the SLEEP state
10	Sleep	RD	0.00896	Node in SLEEP mode
11	Wake-up prescan	1910	1.13	Wake-up for Rx
12	Listen	AT	8.68	Actively Listen for incoming DL packet
13	Scan message	1160 (or 900)	9.64	Rx 24 B FQ data (or 22 B FU)
14	Radio off	389	4.69	Main radio in OFF state
15	Post-processing and Cool down	23,410	0.00619	Set up the sleep timer for the next state and switch to SLEEP state

5. System Model

This section presents the analytical model of the system described in Section 4. The DL latency, FQ occupancy, and PDR are calculated. All the symbols used in this section are summarized in Appendix A.

5.1. The Queuing Model

We call the time interval between two FPs, when the FQ at the start of the interval is not empty, a 'service'. The time interval between two FPs, during which an FU is transmitted from the FN to the LPN (i.e., the FQ at the start of the interval is empty), is called a 'vacation'. The system can be modelled as a finite capacity queue, where the server takes repeated vacations. The duration of a service and a vacation depends on the LPN's capacitor voltage. Let the packets arrive at the FN according to a Poisson distribution with data arrival rate λ , and the FN temporarily buffers them in a buffer (FQ) of capacity N . The vector \mathbf{U} represents the discretized voltages that a capacitor can take between 1 and V_{max} . Hence, the duration of a service when the voltage equals $i \in \mathbf{U}$ can be denoted by S_i with distribution function $\mathbb{S}_i(t)$ and Laplace Transform (LST) $\mathbb{S}_i^*(\nu)$. Similarly, for the vacation, W_i , $\mathbb{W}_i(t)$ and $\mathbb{W}_i^*(\nu)$ can represent the duration, the distribution function, and its LST, respectively. The duration of a service consists of all the radio states during polling and possibly followed by a sleep period to reach the threshold index V_{th}^{FP} . The radio states during the polling can be assumed to have a constant duration for a fixed size of the DL packet. Hence, S_i can be computed considering FQ is not empty and W_i considering it is empty, as discussed in Section 4.2. If at the start of a service the voltage is equal to i , then the voltage at the next poll instance is denoted by \mathbb{V}_i^S and calculated using Equation (1) for all the polling radio states. Similarly, if the voltage is equal to i at the start of a vacation, its values at the next inspection instant denoted by \mathbb{V}_i^W also include the SLEEP state for T_{PI} .

The probability $\mathbb{P}_{k,i}^S$ and $\mathbb{P}_{k,i}^W$ that k packets arrive during the service time and the vacation time respectively, starting from \mathbf{U} , equal to i , can be calculated as Equation (6).

$$\begin{aligned} \mathbb{P}_{k,i}^S &= \int_0^\infty \frac{(\lambda t)^k}{k!} e^{-\lambda t} d\mathbb{S}_i(t) \\ \mathbb{P}_{k,i}^W &= \int_0^\infty \frac{(\lambda t)^k}{k!} e^{-\lambda t} d\mathbb{W}_i(t) \end{aligned} \quad (6)$$

The column vectors of the averages of the service times and the vacation times are given by Equation (7).

$$\begin{aligned} \overline{\mathbf{E}[S]} &= (\mathbf{E}[S_1], \mathbf{E}[S_2], \dots, \mathbf{E}[S_{V_{max}}])' \\ \overline{\mathbf{E}[W]} &= (\mathbf{E}[W_1], \mathbf{E}[W_2], \dots, \mathbf{E}[W_{V_{max}}])' \end{aligned} \tag{7}$$

5.2. Embedded Markov Chain

The system is observed at the poll instants $(t_n, n = 1, 2, \dots, \infty)$. We refer to these poll instants as *inspection instants*. Let Q_n be the number of packets in the FQ and V_n be the available capacitor voltage at time instant t_n . It should be noted that V_n will never take values below $V_{startpoll}$, which is the minimum voltage available at a poll instant that allows all functions of the radio to receive a packet. Hence, the stochastic process (Q_n, V_n) at t_n is a discrete-time finite Markov Chain with state-space size equal to $(N + 1) \cdot (V_{max} - V_{startpoll} + 1)$. The limiting probability distribution function of this Markov Chain is denoted by $\bar{p} = (\bar{p}_0, \bar{p}_1, \dots, \bar{p}_N)$. Let $(\bar{p}_k)_i$ denote the joint probability distribution of the voltage and the queue length, as shown in Equation (8).

$$\begin{aligned} (\bar{p}_k)_i &= \lim_{n \rightarrow \infty} Prob(Q_n = k, V_n = i), \\ &\text{where, } 0 \leq k \leq N, i = V_{startpoll}, \dots, V_{max} \end{aligned} \tag{8}$$

The transition matrix of this Markov Chain is given by Equation (9).

$$\mathbf{P} = \begin{bmatrix} B_0 & B_1 & B_2 & \dots & B_{N-2} & B_{N-1} & \sum_{n=N}^{\infty} B_n \\ A_0 & A_1 & A_2 & \dots & A_{N-2} & \sum_{n=N-1}^{\infty} A_n & 0 \\ 0 & A_0 & A_1 & \dots & A_{N-3} & \sum_{n=N-2}^{\infty} A_n & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & A_0 & \sum_{n=1}^{\infty} A_n & 0 \\ 0 & 0 & 0 & \dots & 0 & \sum_{n=0}^{\infty} A_n & 0 \end{bmatrix} \tag{9}$$

where, B_n represents the transitions starting from an empty system and ending up with n packets in the FQ at the next inspection point. Whereas, A_n represents the transitions starting from a non-empty system. The $(V_{max} - V_{startpoll} + 1) \cdot (V_{max} - V_{startpoll} + 1)$ size-matrices A_n are defined by Equation (10).

$$(A_n)_{i,j} = \begin{cases} \mathbb{P}_{n,i}^S, & \text{for } i = [V_{startpoll}, V_{max}], j = \mathbb{V}_i^S \\ 0, & \text{elsewhere} \end{cases} \tag{10}$$

Similarly, B_n can also be calculated as Equation (11).

$$(B_n)_{i,j} = \begin{cases} \mathbb{P}_{n,i}^W, & \text{for } i = [V_{startpoll}, V_{max}], j = \mathbb{V}_i^W \\ 0, & \text{elsewhere} \end{cases} \tag{11}$$

In the numerical examples, for a large value of k , \mathbf{P}^k has positive real entries. Therefore, the matrix \mathbf{P} is regular and has a unique normalized eigenvector \bar{p} corresponding to the eigenvalue 1. Hence, the probability distribution satisfies Equation (12).

$$\begin{aligned} \bar{p} &= \bar{p} \cdot P \\ \bar{p} \cdot \bar{e} &= 1, \end{aligned} \tag{12}$$

where, \bar{e} is the $(N+1) \cdot (V_{max} - V_{startpoll} + 1)$ unit vector.

The steady-state probability of the system occupancy at arbitrary time instants has been derived in [20]. Let T_{FP} be the average time between two consecutive inspection instants given by Equation (13).

$$T_{FP} = [1 - \bar{p}_0 \cdot \bar{e}] \cdot \mathbf{E}[S] + \bar{p}_0 \cdot \bar{e} \cdot \mathbf{E}[W], \quad (13)$$

where

$$\mathbf{E}[S] = \begin{cases} \frac{\sum_{n=1}^N \bar{p}_n \cdot \overline{\mathbf{E}[S]}}{1 - \bar{p}_0 \cdot \bar{e}}, & \text{if } \bar{p}_0 \cdot \bar{e} \neq 1 \\ 0, & \text{if } \bar{p}_0 \cdot \bar{e} = 1 \text{ (system is always empty)} \end{cases} \quad (14)$$

and

$$\mathbf{E}[W] = \begin{cases} \frac{\bar{p}_0 \cdot \overline{\mathbf{E}[W]}}{\bar{p}_0 \cdot \bar{e}}, & \text{if } \bar{p}_0 \cdot \bar{e} \neq 0 \\ 0, & \text{if } \bar{p}_0 \cdot \bar{e} = 0 \text{ (system is never empty)} \end{cases} \quad (15)$$

The queue occupancy distribution, i.e., the probability that there are n packets in the FQ, is then given by Equation (16) [20].

$$O_n = \frac{1}{\lambda \cdot T_{FP}} \cdot [\bar{p}_n \cdot \bar{e} - \bar{p}_0 \cdot B_n \cdot \bar{e}], \quad \text{where, } n = [0, N-1]$$

And

$$O_N = 1 - \frac{1}{\lambda \cdot T_{FP}} \cdot [1 - \bar{p}_0 \cdot \bar{e}]. \quad (16)$$

Therefore, the total percentage of packet loss is equal to $100 \cdot O_N$.

5.3. Performance Metrics

In this section, the metrics, average FQ occupancy, and average DL latency, are computed.

- Average FQ occupancy: The FN uses a push-out strategy to manage its FQ, which means when a packet arrives at a full FQ, the oldest packet in the FQ would be dropped. However, the number of packets in the *push-out* system is the same even if the arriving packet is dropped (drop-tail). Since the probability that an arriving packet finds a full FQ is equal to O_N , the average system occupancy in a 'drop-tail' system is given by Equation (17).

$$\mathbf{O}_{DT} = \sum_{n=1}^N n \cdot O_n. \quad (17)$$

Now, the average system occupancy in a 'push-out' system (\mathbf{O}_{PO}) is equal to the \mathbf{O}_{DT} .

- Average DL latency: The actual arrival rate in a drop-tail system is given by $(1 - O_N) \cdot \lambda$ and, applying Little's well-known formula [21], the average response time in a drop-tail system is given by Equation (18).

$$\mathbf{T}_{DT} = \frac{\mathbf{O}_{DT}}{(1 - O_N) \cdot \lambda}. \quad (18)$$

However, the waiting time of a packet in the system using push-out is quite different from the waiting time in a system using drop-tail. In a drop-tail system, the packets behind a tagged packet do not impact the delay of that packet. However, in a push-out system, packets arriving after the tagged packet may, in case they arrive at a full queue, push out the first packet in the queue and hence influence the delay experienced by the tagged packet. The waiting time in the push-out system consists of two parts: firstly, the time interval between the packet arrival instant (at FQ) and the next inspection instant (i.e., until the end of the vacation if it arrives during a vacation or until the

end of a service if it arrives during a service); secondly, the time interval between this inspection instant and the start of the transmission of the packet.

- Between arrival instant and the next inspection instant: There are two scenarios, either the packet arrives during the vacation period or during the service period.
 - * An arbitrary time instant falls in a vacation: Let $P_n^\omega(t)$ be the probability that an arbitrary time instant falls in a vacation period, that there are n packets (maximum N) present in the system at time t , and that the remaining vacation time is T_v that satisfies $t \leq T_v < t + dt$. The LST of this probability has already been determined by Blondia [20]. Hence, the inverse LST can provide us the value of $P_n^\omega(t)$ and $P_N^\omega(t)$, as defined in Equation (19).

$$P_n^\omega(t) = \frac{1}{\lambda \cdot T_{FP}} \cdot \left[\bar{p}_0 \cdot \bar{e} \cdot \mu_{E[W]}(t) \cdot (-1)^{n+1} \cdot e^{\lambda \cdot (t - E[W])} \cdot \frac{(t - E[W])^n}{n!} - \sum_{l=0}^n (\bar{p}_0 \cdot B_l \cdot \bar{e} \cdot \lambda^{n-l+1} \cdot (-1)^{n-l+1} \cdot e^{\lambda \cdot t} \cdot \frac{t^{n-l}}{(n-l)!}) \right]. \tag{19}$$

and

$$P_N^\omega(t) = \frac{\bar{p}_0 \cdot \bar{e}}{T_{FP}} \cdot [1 - \mu_{E[W]}(t)] - \sum_{n=0}^{N-1} P_n^\omega(t),$$

where, $n = [0, N - 1]$ and $\mu_{E[W]}(t)$ is a Heaviside function that outputs (0) if $t \leq E[W]$ and (1) if $t > E[W]$.

$P_{n,k}^\omega(t)$ represents the probability that an arbitrary time instant falls in a vacation period, that there are n packets (maximum N) present in the system at time t , and there are k packet arrivals (maximum $N - 1$) during the remaining vacation time T_v . This probability is given by Equation (20).

$$P_{n,k}^\omega(t) = e^{-\lambda \cdot t} \frac{(\lambda \cdot t)^k}{k!} \cdot P_n^\omega(t). \tag{20}$$

The probability that at an arbitrary time instant there are n packets (maximum $N - 1$) in the system and that, during the remaining time of the vacation k , packets (maximum $N - 1$) have arrived is given in Equation (21).

$$P_{n,k}^\Omega = \int_0^\infty P_{n,k}^\omega(t) dt = \frac{1}{\lambda T_{FP}} \sum_{l=0}^n (\bar{p}_0 \cdot B_l \cdot \bar{e}) \cdot \left[\frac{(\lambda \cdot E[W])^{n-l+k+1} \cdot (-1)^{n-l}}{(n-l)! \cdot k! \cdot (n-l+k+1)} \right] \tag{21}$$

And

$$P_{N,k}^\Omega = \frac{\bar{p}_0 \cdot \bar{e}}{\lambda T_{FP}} \cdot \left[1 - e^{-\lambda \cdot E[W]} \cdot \sum_{i=0}^k \frac{(\lambda \cdot E[W])^i}{(i)!} \right] - \sum_{n=0}^{N-1} P_{n,k}^\Omega.$$

The average waiting time until the next inspection instant of a packet that arrives at an arbitrary time instant in a vacation period is then given by Equation (22).

$$T_{n,k}^\Omega = \int_0^\infty t \cdot P_{n,k}^\omega(t) dt = \frac{1}{\lambda T_{FP}} \sum_{l=0}^n (\bar{p}_0 \cdot B_l \cdot \bar{e}) \cdot \left[\frac{(\lambda \cdot \mathbf{E}[W])^{n-l+k+1} \cdot (-1)^{n-l} \cdot \mathbf{E}[W]}{(n-l)! \cdot k! \cdot (n-l+k+2)} \right] \tag{22}$$

And

$$T_{N,k}^\Omega = \frac{\bar{p}_0 \cdot \bar{e} \cdot (k+1)}{\lambda^2 \cdot T_{FP}} \cdot \left[1 - e^{-\lambda \cdot \mathbf{E}[W]} \cdot \sum_{i=0}^{k+1} \frac{(\lambda \cdot \mathbf{E}[W])^i}{(i)!} \right] - \sum_{n=0}^{N-1} T_{n,k}^\Omega.$$

- * An arbitrary time instant falls in a service: Let $P_n^\pi(t)$ be the probability that an arbitrary time instant falls in a service period, that there are n packets (maximum N) present in the system at time t , and that the remaining service time T_s satisfies $t \leq T_s < t + dt$. Based on the Laplace transform, [20] $P_n^\pi(t)$ can be calculated as Equation (23).

$$P_n^\pi(t) = \frac{1}{\lambda \cdot T_{FP}} \cdot \left[\sum_{l=1}^n (\bar{p}_1 \cdot \bar{e} \cdot \lambda^{n-l} \cdot \mu_{\mathbf{E}[S]}(t) \cdot (-1)^{n-l+1} \cdot e^{\lambda \cdot (t-\mathbf{E}[S])} \cdot \frac{(t-\mathbf{E}[S])^{n-l}}{(n-l)!}) - \sum_{l=1}^n (\bar{p}_{l-1} - \bar{p}_0 \cdot B_{l-1} \cdot \bar{e} \cdot \lambda^{n-l} \cdot (-1)^{n-l+1} \cdot e^{-\lambda \cdot t} \cdot \frac{t^{n-l}}{(n-l)!}) \right], \tag{23}$$

and

$$P_N^\pi(t) = \frac{\sum_{l=1}^{N-1} \bar{p}_1 \cdot \bar{e}}{T_{FP}} \cdot \left[1 - \mu_{\mathbf{E}[S]}(t) \right] - \sum_{n=1}^{N-1} P_n^\pi(t).$$

where, $n = [0, \dots, N-1]$.

The probability that a packet arrives in a service period and n packets are present in the FQ at time t and k packets arrive in the remaining service time is given by Equation (24).

$$P_{n,k}^\pi(t) = e^{-\lambda t} \frac{(\lambda t)^k}{k!} \cdot P_n^\pi(t) \tag{24}$$

where, $n = [0, N]$ and $k = [0, N-1]$.

Similar to Equation (21), the probability ($P_{n,k}^\Pi$) that, at an arbitrary time instant, there are n packets in the system and that during the remaining time of the service k packets have arrived can be calculated as Equation (25).

$$P_{n,k}^\Pi = \int_0^\infty P_{n,k}^\pi(t) dt = \frac{1}{\lambda T_{FP}} \sum_{l=1}^n (\bar{p}_{l-1} - \bar{p}_0 \cdot B_{l-1} \cdot \bar{e}) \cdot \left[\frac{(\lambda \cdot \mathbf{E}[S])^{n-l+k+1} \cdot (-1)^{n-l}}{(n-l)! \cdot k! \cdot (n-l+k+1)} \right]. \tag{25}$$

and

$$P_{N,k}^\Pi = \frac{\sum_{l=1}^N \bar{p}_l \cdot \bar{e}}{\lambda T_{FP}} \cdot \left[1 - e^{-\lambda \cdot \mathbf{E}[S]} \cdot \sum_{i=0}^k \frac{(\lambda \cdot \mathbf{E}[S])^i}{(i)!} \right] - \sum_{n=1}^{N-1} P_{n,k}^\pi.$$

The average waiting time of a packet that arrives at an arbitrary time instant in a service period is then given by Equation (26).

$$T_{n,k}^\Pi = \int_0^\infty t \cdot P_{n,k}^\pi(t) dt = \frac{\sum_{l=1}^N (\bar{p}_l \cdot \bar{e})}{\lambda T_{FP}} \cdot \left[\frac{(\lambda \cdot \mathbf{E}[S])^{n-l+k+1} \cdot (-1)^{n-l} \cdot \mathbf{E}[S]}{(n-l)! \cdot k! \cdot (n-l+k+2)} \right] \tag{26}$$

and

$$T_{N,k}^\pi = \frac{\sum_{l=1}^N \bar{p}_l \cdot \bar{e} \cdot (k+1)}{\lambda^2 \cdot T_{FP}} \cdot \left[1 - e^{-\lambda \cdot \mathbf{E}[S]} \cdot \sum_{i=0}^{k+1} \frac{(\lambda \cdot \mathbf{E}[S])^i}{(i)!} \right] - \sum_{n=0}^{N-1} T_{n,k}^\pi$$

- Between this inspection instant and the start of the transmission of the packet: Let $T_{n,k}^s$ be the mean waiting time of a packet in the FQ at the *end* of a service that will be served and that has n packets (maximum $N - 2$) ahead and k packets (maximum $N - n - 2$) behind is given by Equation (27).

$$\begin{aligned} T_{0,k}^s &= 0, & \text{for } k &= [0, N - 2] \\ T_{1,k}^s &= \sum_{j=0}^{N-k-2} \left(T_{0,k+j}^s + \mathbf{E}[S] \cdot \frac{(\lambda \cdot \mathbf{E}[S])^j}{j!} \cdot e^{-\lambda \cdot \mathbf{E}[S]} \right), & \text{for } k &= [0, N - 3] \\ T_{n,k}^s &= \sum_{j=0}^{N-n-k-1} \left(T_{n-1,k+j}^s + \mathbf{E}[S] \cdot \frac{(\lambda \cdot \mathbf{E}[S])^j}{j!} \cdot e^{-\lambda \cdot \mathbf{E}[S]} \right) + \\ & \sum_{j=N-n-k}^{N-k-2} \left(T_{N-k-2-j,k+j}^s + \mathbf{E}[S] \cdot \frac{(\lambda \cdot \mathbf{E}[S])^j}{j!} \cdot e^{-\lambda \cdot \mathbf{E}[S]} \right), & \text{for } k &= [0, N - n - 2] \end{aligned} \tag{27}$$

and $n = [2, N - 2]$.

Similar to Equation (27), the mean waiting time of a packet in the FQ at the end of a vacation period $T_{n,k}^v$ that will be served and has n packets ahead (maximum $N - 1$) and k packets (maximum $N - n - 1$) behind can be calculated.

The total waiting time of a packet arriving at an arbitrary instant at the FN or average DL latency, served eventually in a system using the push-out strategy, is given by Equation (28).

$$\begin{aligned} \mathbf{T}_{PO} &= \sum_{k=0}^{N-1} \left(T_{0,k}^\Omega + P_{0,k}^\Omega \cdot T_{0,k}^v \right) + \sum_{n=1}^{N-1} \left[\sum_{k=0}^{N-n-1} \left(T_{n,k}^\Omega + P_{n,k}^\Omega \cdot T_{n,k}^v \right) + \right. \\ & \left. \sum_{k=N-n}^{N-1} \left(T_{N,k}^\Omega + P_{N,k}^\Omega \cdot T_{N-k-1,k}^v \right) \right] + \sum_{k=0}^{N-2} \left(T_{1,k}^\Pi + P_{1,k}^\Pi \cdot T_{0,k}^s \right) + \\ & \sum_{n=2}^{N-1} \left[\sum_{k=0}^{N-n-1} \left(T_{n,k}^\Pi + P_{n,k}^\Pi \cdot T_{n-1,k}^s \right) + \sum_{k=N-n}^{N-2} \left(T_{N,k}^\Pi + P_{N,k}^\Pi \cdot T_{N-k-2,k}^s \right) \right] \end{aligned} \tag{28}$$

6. Evaluation

This section presents the evaluation of the DL latency and PDR results using the analytical model and the simulator. Firstly, the description of the simulation setup is presented. This is followed by comparing the results of our analytical model and the simulator. The model is implemented in Matlab R2020b, while the simulation program is written in Python 3.9.6. Thereafter, these results are analyzed in detail based on parameters

such as harvesting power, capacitor size, data arrival rate, and poll interval timer of a batteryless BLE LPN.

Note that the execution time of the model in Matlab scales with the size of the transition matrix. Therefore, if it is known that, at a specified harvesting rate, the capacitor would be unable to recharge itself up to V_{max} and the matrix size can be reduced by assuming a lower value of V_{max} . This behaviour can be easily perceivable for low harvesting rates in combination with large capacitor sizes.

6.1. Simulation Setup

We implemented an event-based simulator to imitate the friendship communication mechanism of the batteryless LPNs as proposed in [7]. The simulator follows an event-based packet scheduling and is capable of reproducing the BLE radio activities, such as sending FP messages and FQ buffered data packets. It also uses the same capacitor model to imitate the capacitor behaviour, as mentioned in Section 4. Each experiment of the simulator is run until a total of 25,000 packets have been generated according to a Poisson arrival process. To request and receive a buffered data packet, the LPN follows the sequence of states listed in Table 1, in the order from top to bottom. The table also mentions the execution time and the current consumption of the corresponding states of the LPNs. These time and current consumption values can be used as Δt and $I_L(X)$ to calculate the capacitor's voltage at any time using Equation (1). The time and the current consumption were measured at 1.8V supply voltage on the Nordic nRF52840 BLE devkit [22] using the Nordic Power Profiler Kit II [23] while running the developed LPN application. All the states of the FP and receiving message mentioned in Table 1 can be seen in Figure 5.

The parameters listed in Table 2 are used to evaluate the model. As such, the values of V_{cutoff} and V_{max} are taken considering one of the configurations of a representative off-the-shelf power management unit [24]. The following performance metrics are considered in the comparison:

- Packet delivery ratio (PDR): The ratio of packets successfully received by the LPN compared to the total reaching the FQ.
- Downlink packet latency: The average latency to receive a packet by the LPN from the time it arrives at the FQ.

Table 2. Simulation parameters.

Parameters	Symbol	Values/Range
Poisson packet arrival rate	μ_p	[1, 10, 30, 60, 120] s
Harvesting power	P_h	[0.1, 0.15, 0.25, 0.3, 0.35, 0.2, 0.4, 0.6, 0.8, 1] mW
Capacitor size	C	[0.1, 0.2, 0.3, 0.4, 0.5, 1] mF
Poll interval	T_{PI}	[1, 10, 20, 40, 60, 80, 100, 150, 200, 250, 300, 400, 500, 600, 800, 1000] s
Turn-off voltage	V_{cutoff}	2.8 V
Max operating voltage	V_{max}	4.5 V
Friend Queue size	N	[8, 16] packets
Receive Delay	T_{RD}	255 ms
Arrival time (max up to T_{RW})	T_{AT}	0 ms
Tx power level		0 dBm

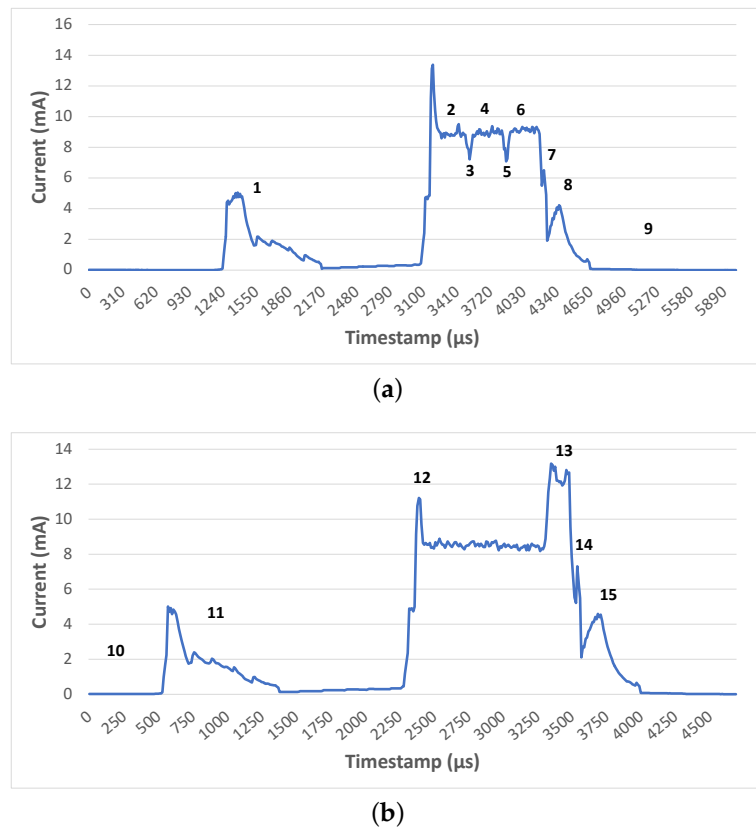


Figure 5. Current consumption of nRF52840 with the state number as defined in Table 1. (a) Current consumption during FP; (b) Current consumption during Scan.

6.2. Model Validation

The model is validated using the simulation and the emulator. We ran a total of 5640 test cases to compare the results of the analytical model and the simulation. The test cases encompass all possible parameter combinations listed in Table 2. Whereas, using an emulator, a hardware with a lack of automation, we manually ran 25 tests.

6.2.1. Using Simulation

The harvesting power is considered for harvesting techniques, which are low indoor light or human body temperature, that generate up to 1 mW. To avoid longer latency delay, the poll interval is considered only up to 10 times the data rate. These test cases only analyze DL data. The cumulative distribution shown in Figure 6 presents the relative deviation in PDR and DL latency between the simulation and the model results. We also simulate the device for the large power (more than 1 mW), generating harvesting techniques such as industrial temperature, direct sunlight, or mechanical movement. However, the result obtained with such harvesting techniques would result in 100% PDR with the lowest possible DL latency. Therefore, we exclude these test cases from the result analysis.

There is a mean difference of 2.23% and 0.099% in DL latency and PDR, respectively, found in a run of all the test cases. A high relative difference in PDR value ($\geq 2\%$) is observed when the PI is set to a relatively larger value than the data arrival rate. For these cases, the relative difference in latency is around 13%. Moreover, the large relative deviation in latency value ($\geq 15\%$) is also observed for the low P_h when the PDR is less than 100%. However, the deviation in latency is higher for the lowest FQ sizes (8 or less). For such cases, the deviation in PDR is only 0.58%.

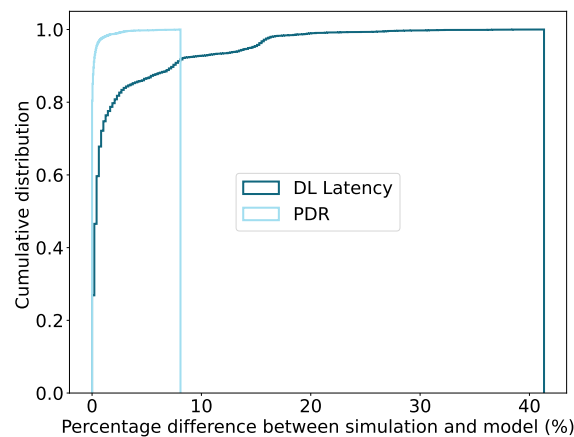


Figure 6. Cumulative distribution of difference in model and simulation results for both the DL latency and PDR metrics.

There are a total of 92.73% and 100% test cases for DL latency and PDR, respectively, with a percentage difference less than 10%. By analyzing the results for the PDR and DL latency of the model and simulation, it is evident that the model's accuracy is high. However, solving the model is computationally complex for large FQ sizes (32 or more), and the relative mean difference in the DL latency test cases for the lowest FQ size (2 packets) is higher compared to larger queue sizes.

6.2.2. Using Emulator

We use an energy-harvesting emulator called Environment Emulator (EnvEmu) [25], which is a stripped-down version of the TMote Sky. It is connected to the Nordic nRF52840 development board that runs the BLE LPN application, as shown in Figure 7. The EnvEmu is powered by USB and provides a variable input voltage between 1.5 and 3.3 V to Nordic BLE LPN via its VDD-GND pin. The output voltage is based on the emulation of a virtual capacitor and energy harvesting source, which can both be dynamically configured in the EnvEmu via simple commands. An instruction manual for the EnvEmu using Linux can be found in [26]. Two more Nordic nRF52832 development boards powered by USB are also part of the BLE mesh network, of which one acts as an FN and another generates DL packets. The FN temporarily accumulates the incoming DL data from the mesh node, which the batteryless LPN polls.

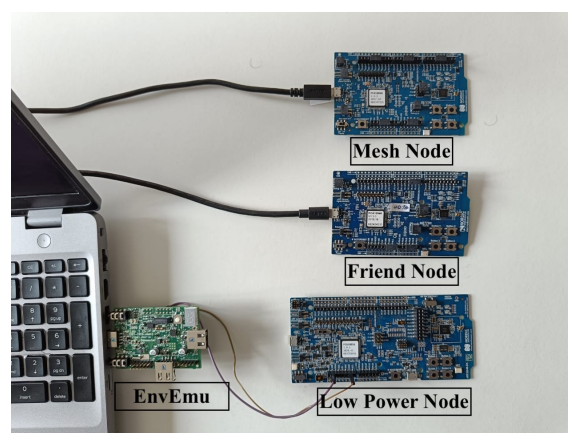


Figure 7. EnvEmu setup.

Figure 8 shows the comparison of the model and emulator for different FQs. It can be observed that the results are similar. The mean difference in the result of the emulator and model is 1.22% and 0.3% in PDR and DL latency, respectively.

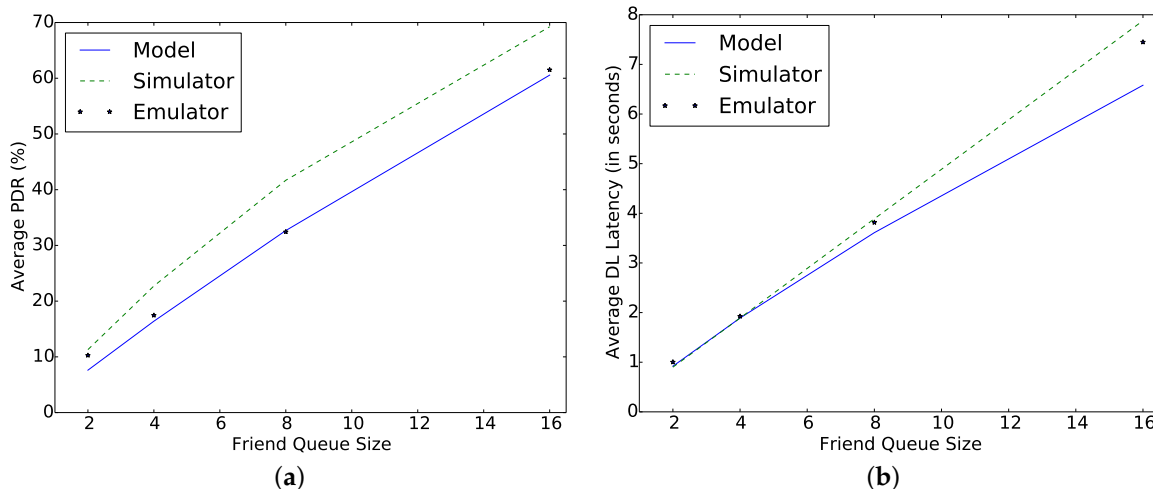


Figure 8. Comparing emulator, simulator, and model for different FQ at $\mu_p = 1$ s, $T_{PI} = 30$ s, $C = 1$ mF, $P_h = 0.1$ mW, $T_{AT} = 5$ ms. (a) Average packet delivery ratio; (b) DL latency.

6.3. Result Analysis Based on Analytical Model and Simulation

Figures 9–11 plot the results of the analytical model and the simulation, showing the effect of all the dependent parameters. It is clear that the analytical model and simulation follow the same behaviour, and the results match closely in all the plotted graphs. It can be observed from Figure 9 that, as the harvesting power increases, the average PDR and DL latency improve until they reach their corresponding maximum achievable values.

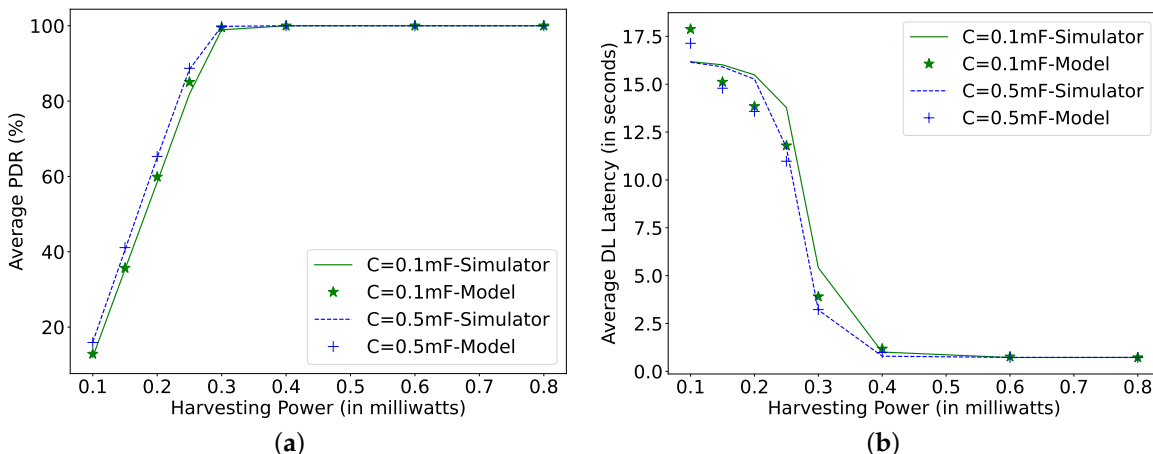


Figure 9. Comparing simulator and model for different harvesting powers at $\mu_p = 1$ s, $T_{PI} = 1$ s, $N = 16$, $T_{AT} = 0$ ms. (a) Average packet delivery ratio; (b) DL latency.

The improvement in the PDR and DL latency accelerate faster for low harvesting power until the PDR crosses 80% (0.25 mW in Figure 9a,b). Further, an increase in the harvesting power improves the DL latency at a higher rate than the PDR because, at such harvesting power, the capacitor can recharge faster again to V_{th}^{FP} to receive packets. As soon as the PDR and DL latency reach their corresponding best achievable values, an increase in the harvesting power does not affect the results because the capacitor voltage never drops below V_{th}^{FP} , and therefore, the node does not need to go into SLEEP mode to recharge its capacitor. For an arrival rate of 1 s, this harvesting power point is 0.6 mW, as shown in Figure 9. With the decrease in arrival rate, the capacitor gets more time to recharge itself. Therefore, for an arrival rate of 10 s, the mentioned harvesting power point value reduces to 0.3 mW (not shown in Figures). The size of the capacitor also plays a vital role

in achieving the desired performance. As a larger capacitor can store more energy, the PDR for the 0.5 mF capacitor is higher than for 0.1 mF. However, the increase of the capacitor size also affects its charging time. Therefore, the performance does not improve much with the increase in the capacitor size after a certain optimal capacitor size for each specific harvesting power. There is an optimal capacitor size for each set of parameters. It can be observed from Figure 10b that, at a harvesting power of 0.1 mW, the optimal capacitor size, where the DL latency is lowest, is 0.35 mF, and for 0.2 mW of harvested power, it is 0.45 mF. An increase in the capacitor size higher than 0.5 mF shows reduced growth in the PDR (harvesting power 0.1 and 0.2 mW). Additionally, when the PDR is 100%, the increase in the capacitor size would improve the DL latency (yellow line at P_h 0.4 mW), but not substantially.

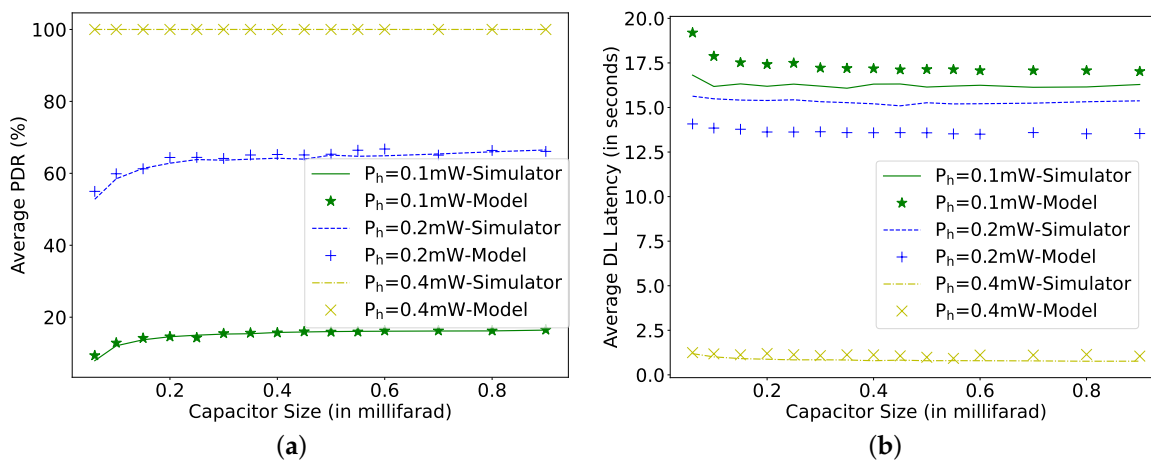


Figure 10. Comparing simulator and model for different capacitor sizes and harvesting power at $\mu_p = 1$ s, $T_{PI} = 1$ s, $N = 16$, $T_{AT} = 0$ ms. (a) Average packet delivery ratio; (b) DL latency.

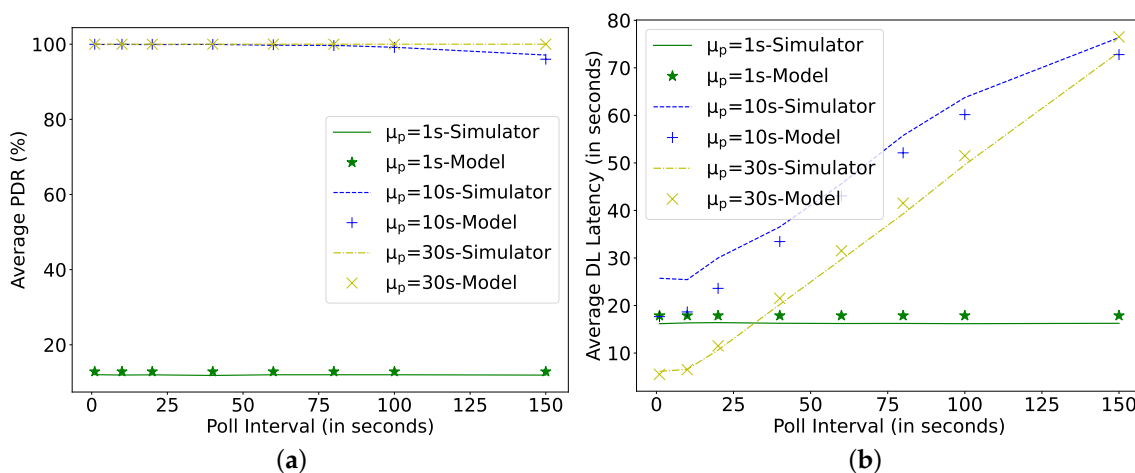


Figure 11. Comparing simulator and model for different Poll Intervals at $C = 0.1$ mF, $P_h = 0.1$ mW, $N = 16$, $T_{AT} = 0$ ms. (a) Average packet delivery ratio; (b) DL latency.

Figure 11 shows that, for a fixed harvesting power (0.1 mW), with the decrease in Poisson arrival rate for downlink packets, the PDR can be improved to 100%. This is because the capacitor gets ample time to charge itself before the subsequent transmission. However, the achieved latency can increase with the increase in the poll interval timer as the node has to wait for this PI once it receives the FU. Therefore, the higher the PI, the longer the arriving packets wait in the FQ to be polled. As such, when the data arrival rate is 10 s, some packets are also being dropped for large PI (150 s). Although frequent

polling consumes additional energy, the considered harvesting power (0.1 mW) is sufficient to support a PI as low as 1 s without negatively affecting PDR. Moreover, the DL latency at the arrival rate of 1 s is not affected by the increase in PI, because there are always packets waiting in the FQ to be polled, and the LPN never receives an FU message.

If the size of the FQ decreases, then the number of dropped packets increases for higher PI. This can be observed in Figure 12. The smaller the FQ, the lower the PDR. Another interesting parameter of the LPN is the AT, which also affects the power consumption and DL latency, as the LPN needs to actively listen during the configured time. With the increase in AT, the LPN wastes energy in idle listening on the incoming channels.

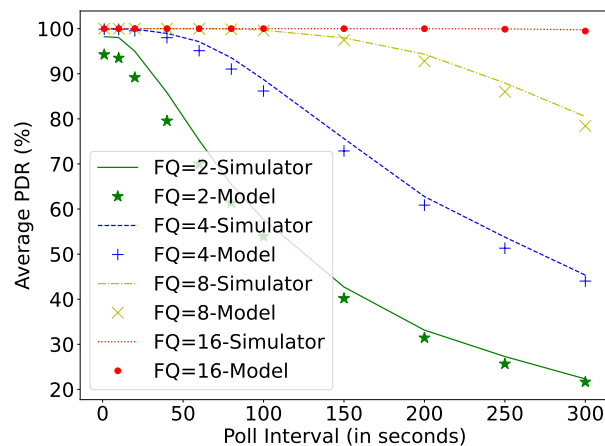


Figure 12. Comparing simulator and model for different poll intervals and FQ at $C = 0.1$ mF, $P_h = 0.1$ mW, $\mu_p = 30$ s, $T_{AT} = 0$ ms.

Figure 13 shows its effect on DL latency and PDR. It can be observed that a slight increase in AT can drastically affect the PDR and, therefore, the DL latency. However, when the data arrival rate is low, the LPN has sufficient time to recharge its capacitor to maintain a PDR of 100%, but the increase in AT affects the DL latency exponentially. As seen for the Poisson arrival rate of 30 s, the latency increases 60-fold from 0.65 s at 0 s AT to 38.43 s at AT of 255 ms. At a high Poisson arrival rate (i.e., 1 s), the impact of the AT is higher because the LPN has less time to charge its capacitor and needs to receive more packets compared to lower Poisson data arrival rates. Increasing the AT further affects the performance marginally. Therefore, AT should be configured as small as possible to achieve high LPN performance. However, a small AT value provides less time to the FN to send a response to the LPN, which in turn affects the scalability (i.e., it causes missed response opportunities when multiple LPNs expect a response simultaneously).

Furthermore, from Figure 14, it can be observed that an LPN having a capacitor of 0.1 mF and a harvesting power of 0.1 mW can receive almost all the data packets when the arrival rate is 10 s for poll interval timers lower than 60 s. Further, decreasing the data arrival rate (from 7 s to 30 s) would improve the latency because fewer packets are entering into the FQ and the capacitor gets more time to recharge itself. Similarly, it improves the PDR when the data arrival rate decreases from 1 s to 7 s. However, the DL latency increases in this interval because with the decrease in data interval fewer packets are entered into the FQ and thus wait longer to be polled.

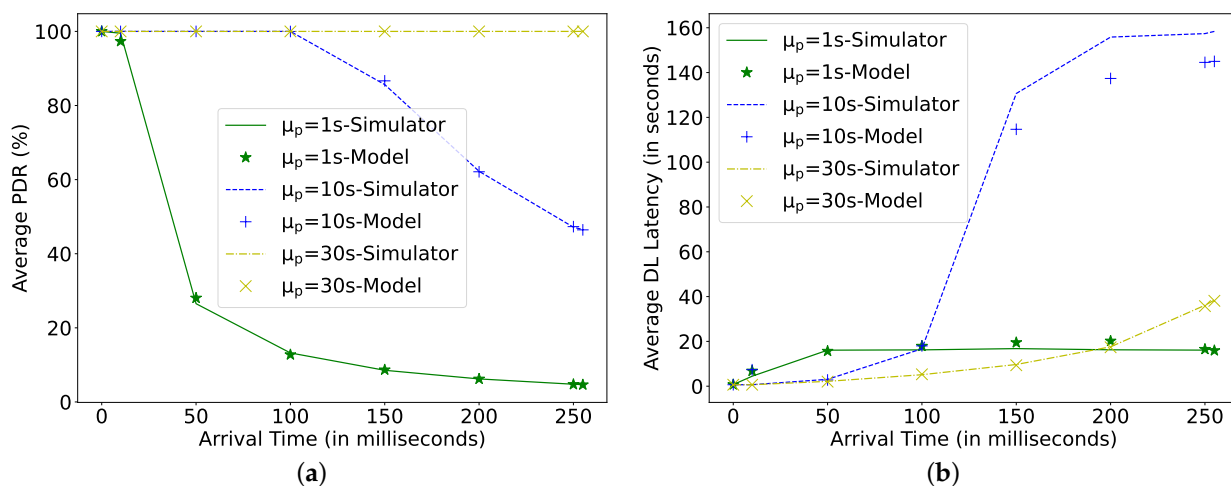


Figure 13. Comparing simulator and model for different arrival times (AT) at C = 4 mF, P_h = 1 mW, N = 16, T_{PI} = 1 s. (a) Average packet delivery ratio; (b) DL latency.

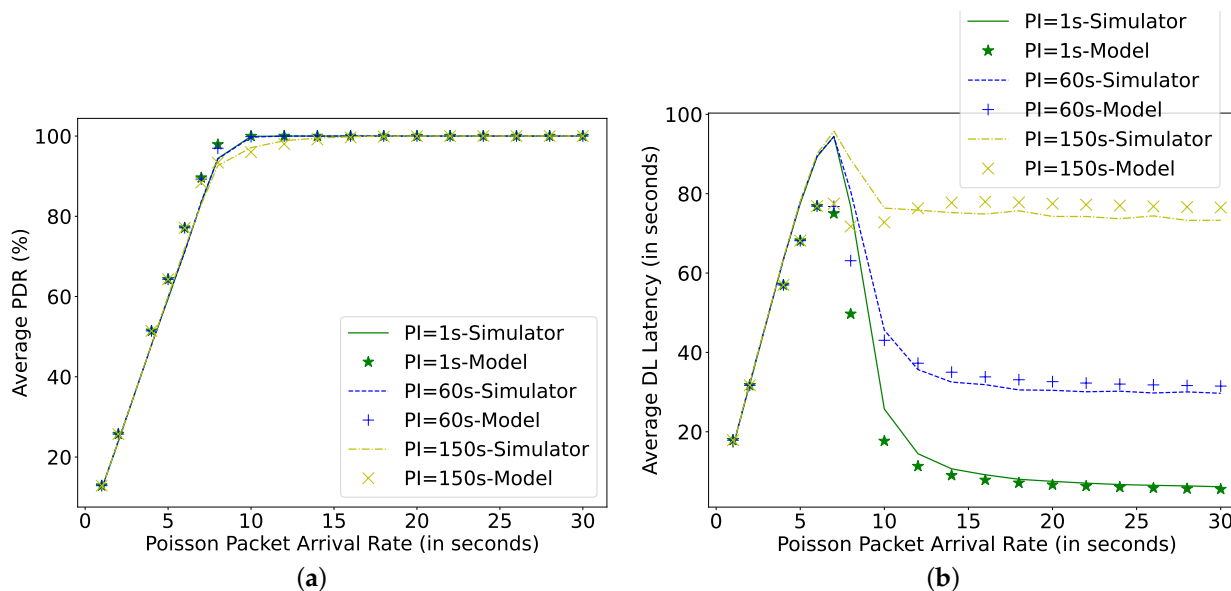


Figure 14. Comparing simulator and model for different packet arrival rate at C = 0.1 mF, P_h = 0.1 mW, N = 16. (a) Average packet delivery ratio; (b) DL latency.

7. Use-Case: Logistic Management

This section presents the use of the proposed model to derive optimal parameters for a batteryless BLE node, considering the IoT use-case of the logistics process. Logistics is one of the major industries adopted by IoT at scale. IoT applications can facilitate asset monitoring in warehouses or shops. Instead of checking the assets manually, they can be connected via batteryless BLE devices that allow effortless tracking of each item. This batteryless device can listen to incoming queries to respond about its availability or transmit sensor data. Consider a use-case where a server queries for the asset availability every 30 s and the batteryless node can harvest 100 μW. The model can be used to find the optimal BLE batteryless LPN configuration, including the capacitor size and poll interval. Assuming a friend queue size of eight packets and an arrival time of 5 min, the model provides the result as shown in Figure 15. It can be observed that the PDR is 100% for lower PI values and the capacitor which are greater than equal to 0.5 mF. However, DL latency is lower for a capacitor of 5 mF or more and does not vary much. Therefore, considering the charging and discharging time of a capacitor, we should select the lowest possible capacitor, which,

for this use-case, can be considered as 5 mF. Additionally, the DL latency is lowest (18.27 s) at a PI equal to 20 s.

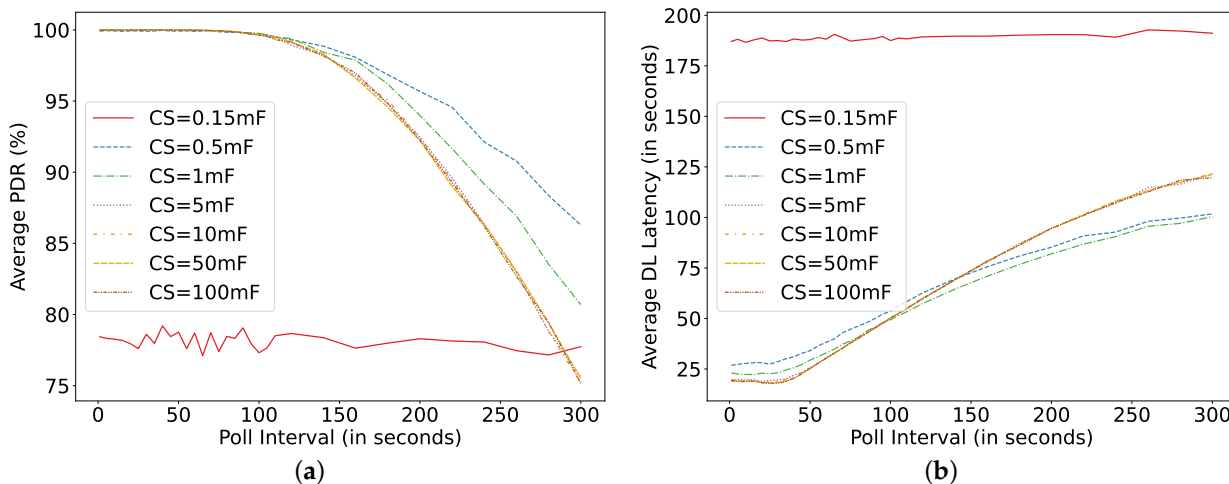


Figure 15. Asset monitoring at DR = 30 s, $P_h = 0.1$ mW, $N = 8$, $T_{AT} = 5$ ms. (a) Average packet delivery ratio; (b) DL latency.

Moreover, if the packet arrival rate from the server changes, the model can be used to determine the new expected performance of this batteryless LPN that harvests 100 μ W and can be configured with a capacitor of 5 mF and polling at 20 s. The result is shown in Figure 16. It can be observed that, at a high arrival rate (1 and 10 s), many packets drop, whereas, lowering the packet arrival rate (60 and 600 s) improves the DL latency up to 55% without affecting the PDR.

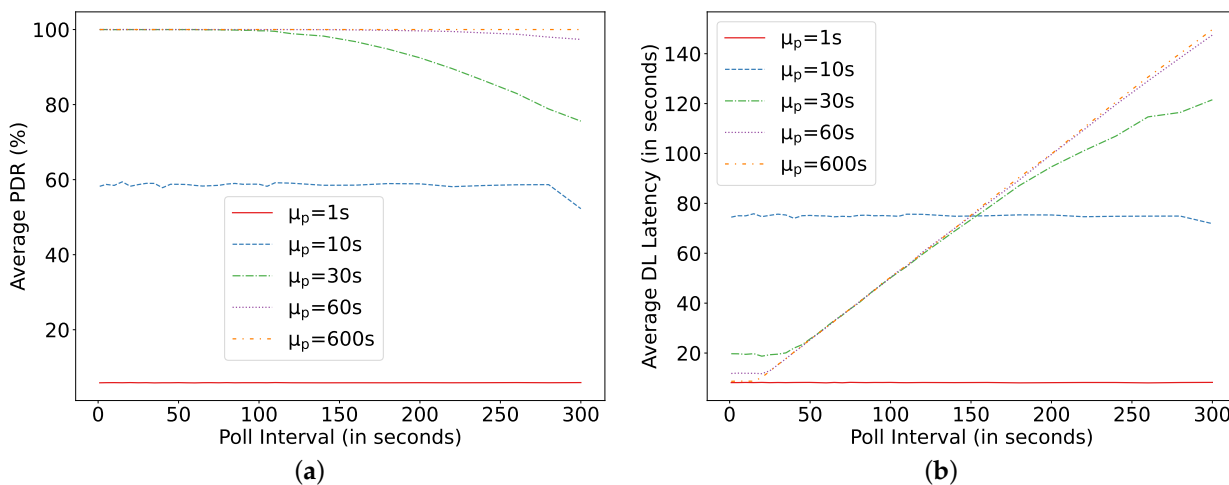


Figure 16. Asset monitoring at C = 5 mF, $P_h = 0.1$ mW, $N = 8$, $T_{AT} = 5$ ms for different packet arrival rate. (a) Average packet delivery ratio; (b) DL latency.

8. Conclusions

In this paper, we presented an analytical model to calculate the DL latency and PDR for a batteryless BLE LPN operating on power harvested from natural or artificial indoor light. The model’s accuracy was compared with Python-based simulation results for various parameters (i.e., capacitor size, harvesting power, data arrival rate, poll interval timer, arrival time, and friend queue size). The results derived from these two methods showed an average deviation of 2.23% in the results of DL latency and 0.09% in PDR. The graphs of different parameters showed diverse behaviour for DL latency and PDR, whereas the model and simulation show the same pattern.

The results showed that a harvesting power of 0.6 mW can maintain a DL packet inter-arrival rate as low as 1 s, and at 0.3 mW, the LPN can receive all the packets arriving at a rate of 10 s to an FQ of size 16. The smallest optimal capacitor size should be chosen depending on the available minimum harvesting power so that it can be recharged rapidly. The poll interval timer affects the DL latency linearly if the PDR is 100%. Therefore, it is important to choose the lowest possible poll interval timer that can be supported by the minimum harvesting power. If the harvesting power varies over time, the poll interval timer should be dynamically adapted. The external factors, such as Friend Queue size, also affect the performance of the batteryless LPN. A small queue size increases the packet loss. In our experiments, a queue size of 16 packets was enough to maintain 100% PDR, for polling intervals as high as 300 s at a downlink packet arrival rate of 30 s. Moreover, the AT parameter should be configured to be as small as possible so that the LPN does not lose energy in idle listening on the incoming channels. However, a very low AT may negatively affect scalability. For a harvesting power of 1mW, an AT up to 100 ms could be supported without a decrease in PDR.

9. Discussion and Future Work

There are several future research directions. In our model, we have considered a power management system that supplies variable voltage directly from the capacitor. However, more advanced power management solutions exist that equalize the supply voltage and only provide the device with power if the capacitor voltage is within certain minimum and maximum bounds [27]. Such solutions offer more consistent and stable performance, as then the power consumption for each radio state would be stable with a stable voltage supply. Most of the existing works in the batteryless domain do not consider this [28–31]. Therefore, in the future, we will explore adapting the batteryless circuit model for such power management solutions. However, in this case, an additional circuit would be required to read the capacitor voltage. CusTARD uses an Analog-to-digital (ADC) pin to read the capacitor voltage [32]. The ADC pin can read values in a range that is usually lower than the capacitor voltage. Therefore, a voltage divider would be required, which increases the power consumption of the device (several tens of μW) and therefore also affects the performance in terms of PDR and DL data latency for different harvesting powers. This could be incorporated into our model by changing Equation (1) and adding the current consumption of the additional circuit to the load current. Other modules used for voltage reading are Torpor Logic and Power switch [33], which can consume up to 2 μW rather than the assumed ultra-low-power comparator that consumes power in the order of pico-watts. As such, we need to perform measurements using real hardware to incorporate this power consumption into the model, as the hardware circuitry to enable measuring the capacitor voltage can impact the results.

The harvesting architecture considered is harvest-store-use. However, there is another architecture ‘harvest-use’ that eliminates the use of capacitors for energy storage [34]. This architecture can reduce the device cost and efficiency. A prototype of such an architecture is presented by Lee et al. [35]. Our model can be extended to a storage-less architecture. In this case, the performance is mainly governed by the harvesting power without a capacitor. Therefore, the stochastic process would consider the number of packets in the FQ and the harvesting source power. A certain minimum harvesting power is then required before and during execution of each task. Therefore, we need another additional circuitry, such as a sense resistor to measure the power of the harvesting source, which further increases the power demand of the batteryless device. Therefore, the requirements and performance of this architecture would be different.

Additionally, we aim to extend the analytical model to consider a variable harvesting power, in contrast to the currently considered constant harvesting power. This would provide a realistic harvesting model to evaluate harvesting sources, such as solar or wind. The model would consider only the current amount of harvested energy. Such solutions are realized by Zhou et al. [36], Patil et al. [37], and Kumar et al. [38] for investigating wireless

sensor networks with energy harvesting. An approach to achieve this is by considering multiple energy chunks that are harvested during different Poisson distributed time slots, where the harvesting power during a slot remains constant [37]. As such, our existing model can be relatively easily extended to such a scenario by applying it to different harvesting power inputs for each variable-length time slot.

Finally, the increased CPU utilization due to servicing multiple LPNs may slow down the LPN's response and thus affect the DL latency. Therefore, another aspect is to study the interactions and effects of multiple LPNs associated with one FN. To date, there are no works that study this aspect. However, research has been done to study the effect on discovery latency due to the number of scanners [39] or due to interference in the presence of many BLE nodes [40]. With an increasing number of BLE devices, delays of device discovery show exponential growth. Due to interference, the latency can increase up to 400% in a network of 30 devices. This implies that severe contention can occur among multiple BLE devices while considering bi-directional communication using a batteryless device.

Author Contributions: Conceptualization, A.K.S., C.D., C.B. and J.F.; methodology, A.K.S. and C.B.; software, A.K.S. and C.B.; validation, A.K.S. and C.D.; formal analysis, A.K.S., C.B. and J.F.; investigation, A.K.S. and C.B.; resources, A.K.S., C.D. and C.B.; data curation, A.K.S. and C.B.; writing—original draft preparation, A.K.S. and C.B.; writing—review and editing, C.D. and J.F.; visualization, C.B. and J.F.; supervision, C.B. and J.F.; project administration, J.F.; funding acquisition, J.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the ICON project BLUESS, the University of Antwerp IOF funded project COMBAT (Time-Sensitive Computing on batteryless IoT Devices), the Flemish FWO SBO S001521N IoBaLeT(Sustainable Internet of batteryless Things) project, and the CERCA Programme from the Generalitat de Catalunya.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Parameters Used in the Model

Table A1 summarizes all the important parameters' symbol with their definition.

Table A1. Parameters Used in the Markov Chain Model.

Symbol	Meaning
λ	Poisson data arrival rate
N	FQ size
\mathbf{U}	Voltage Vector
S_i	Duration of service when $\mathbf{U} = i$
W_i	Duration of vacation when $\mathbf{U} = i$
V_i^S	Voltage at next poll instance when $\mathbf{U} = i$ at the start of a service
V_i^W	Voltage at next poll instance when $\mathbf{U} = i$ at the start of a vacation
$E[S_i]$	Average duration of S_i
$E[W_i]$	Average duration of W_i
\bar{p}	Probability distribution vector of Markov Chain

Table A1. Cont.

Symbol	Meaning
\bar{p}_i	Probability distribution of the voltage equals i
$(\bar{p}_k)_i$	Probability distribution of the voltage equals i if the queue length is k
B_i	Transitions starting from an empty system and ending up with i packets in FQ at the next FP
A_i	Transitions starting from a non-empty system and ending up with i packets in FQ at the next FP
O_n	System occupancy when n number of packets are present in FQ
O_{DT}	Average system occupancy in drop-tail system
O_{PO}	Average system occupancy in push-out system
$\mathbb{P}_{k,i}^S$	Probability that k packets arrive at service time starting from $U = i$
$\mathbb{P}_{k,i}^W$	Probability that k packets arrive at vacation time starting from $U = i$
$P_n^\omega(t)$	Probability that the packet arrives in a vacation period and n packets are present in the FQ at time t
$P_{n,k}^\omega(t)$	Probability that there are packet arrives in a vacation period and n packets are present in the FQ at time t and k packets arrive in the remaining vacation time
$P_{n,k}^\Omega$	Probability that the packet arrives in a vacation period and n packets are present in the FQ and k packets arrive in the remaining vacation time
$P_n^\pi(t)$	Probability that the packet arrives in a service period and n packets are present in the FQ at time t
$P_{n,k}^\pi(t)$	Probability that the packet arrives in a service period and n packets are present in the FQ at time t and k packets arrive in the remaining service time
$P_{n,k}^\Pi$	Probability that the packet arrives in a service period and n packets are present in the FQ and k packets arrive in the remaining service time
$\mu_{E[W]}(t)$	Heaviside function for $E[W]$
$\mu_{E[S]}(t)$	Heaviside function for $E[S]$
T_{FP}	Average Time between two consecutive FPs
T_{DT}	DL latency in drop-tail system
T_{PO}	DL latency in push-out system
$T_{n,k}^\Omega$	Average waiting time of a packet that arrives in a vacation period
$T_{n,k}^\Pi$	Average waiting time of a packet that arrives in a service period
$T_{n,k}^s$	Mean waiting time of a future served packet in the FQ at the end of a service period that has n packets ahead and k packets behind it
$T_{n,k}^v$	Mean waiting time of a future served packet in the FQ at the end of a vacation period that has n packets ahead and k packets behind it

References

- Buchmann, I. *Batteries in a Portable World: A Handbook on Rechargeable Batteries for Non-Engineers*; Cadex Electronics Inc.: Richmond, BC, Canada, 2001.
- Batteries vs. Ultracapacitors? The Answer Is Both. Available online: <https://batteryindustry.tech/batteries-vs-ultracapacitors-the-answer-is-both/> (accessed on 22 January 2022).
- Wang, Z.L.; Lin, L.; Chen, J.; Niu, S.; Zi, Y. *Triboelectric Nanogenerators Green Energy and Technology*; Springer: Berlin/Heidelberg, Germany, 2016.
- Enterprise Bluetooth Applications, Rigado. Available online: <https://www.rigado.com/enterprise-bluetooth-applications/> (accessed on 22 January 2022).
- RC Charging Circuit, Electronics Tutorials. Available online: https://www.electronics-tutorials.ws/rc/rc_1.html (accessed on 22 January 2022).
- Mesh Profile Bluetooth Specifications. Available online: <https://www.bluetooth.com/specifications/mesh-specifications/> (accessed on 22 January 2022).
- Sultania, A.K.; Delgado, C.; Famaey, J. Enabling Low-Latency Bluetooth Low Energy on Energy Harvesting Batteryless Devices Using Wake-Up Radios. *Sensors* **2020**, *20*, 5196. [CrossRef] [PubMed]
- Silva, S.; Martins, H.; Valente, A.; Soares, S. A Bluetooth Approach to Diabetes Sensing on Ambient Assisted Living systems. *Procedia Comput. Sci.* **2012**, *14*, 181–188. [CrossRef]

9. Liu, Q.; IJntema, W.; Drif, A.; Pawełczak, P.; Zuniga M.; Yıldırım, K. S. Perpetual Bluetooth Communications for the IoT. *IEEE Sens. J.* **2021**, *21*, 829–837. [CrossRef]
10. Fraternali, F.; Balaji, B.; Agarwal, Y.; Benini, L.; Gupta, R. Pible: Battery-free mote for perpetual indoor ble applications: Demo abstract. In Proceedings of the 5th Conference on Systems for Built Environments (BuildSys '18), Shenzhen, China, 7–8 November 2018; pp. 168–171.
11. Nilsson, M.; Deknache, H. Investigation of Bluetooth Mesh and Long Range for IoT Wearables. Ph.D. Thesis, Malmö Universitet/Teknik och Samhälle, Malmö, Sweden, 2018.
12. Álvarez, F.; Almon, L.; Hahn, A.; Hollick, M. Toxic Friends in Your Network: Breaking the Bluetooth Mesh Friendship Concept. In Proceedings of the 5th ACM Workshop on Security Standardisation Research Workshop (SSR'19), London, UK, 11 November 2019; pp. 1–12.
13. Darroudi, S.; Caldera-Sánchez, R.; Gomez, C. Bluetooth Mesh Energy Consumption: A Model. *Sensors* **2019**, *19*, 1238. [CrossRef] [PubMed]
14. Álvarez, F.; Almon, L.; Radtke, H.; Hollick, M. Bluemergency: Mediating Post-disaster Communication Systems using the Internet of Things and Bluetooth Mesh. In Proceedings of the 9th IEEE Global Humanitarian Technology Conference, Seattle, WA, USA, 17–20 October 2019.
15. Designing with Bluetooth Mesh: Nodes and Feature Types. Available online: <https://www.embedded.com/designing-with-bluetooth-mesh-nodes-and-feature-types/> (accessed on 22 January 2022).
16. Hortelano, D.; Olivares, T.; Carmen Ruiz, M. Reducing the energy consumption of the friendship mechanism in Bluetooth mesh. *Comput. Netw.* **2021**, *195*, 108172. [CrossRef]
17. Aijaz, A. Infrastructure-less Wireless Connectivity for Mobile Robotic Systems in Logistics: Why Bluetooth Mesh Networking is Important? *arXiv* **2021**, arXiv:2107.05563.
18. Delgado, C.; Sanz, J.M.; Famaey, J. On the Feasibility of Batteryless LoRaWAN Communications using Energy Harvesting. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
19. Gawhare, S.B.; Gaikwad, A. Area and Power efficient High Speed Voltage Comparator. In Proceedings of the International Conference on Automatic Control and Dynamic Optimization Techniques, Pune, India, 9–10 September 2016; pp. 198–201.
20. Blondia, C. A queueing model for a wireless sensor node using energy harvesting. *Telecommun. Syst.* **2021**, *77*, 335–349. [CrossRef]
21. Stochastic, Karl Sigman. Available online: <http://www.columbia.edu/~ks20/stochastic-I/stochastic-I-LL.pdf> (accessed on 4 January 2022).
22. nRF52840 DK, Nordic Semiconductor. Available online: <https://www.nordicsemi.com/Software-and-tools/Development-Kits/nRF52840-DK> (accessed on 22 January 2022).
23. Power Profiler Kit II, Nordic Semiconductor. Available online: <https://www.nordicsemi.com/Software-and-tools/Development-Tools/Power-Profiler-Kit-2> (accessed on 22 January 2022).
24. User Guide Evaluation Board for AEM10941, Epeas. Available online: https://e-peas.com/wp-content/uploads/2020/06/epeas_Solar_Energy_Harvesting_User_Guide_AEM10941.pdf (accessed on 22 January 2022).
25. w-iLab.1 Hardware. Available online: <https://doc.ilabt.imec.be/ilabt/wilab/hardware.html> (accessed on 4 March 2022).
26. Using the Sensor Nodes. Available online: <https://doc.ilabt.imec.be/ilabt/wilab/tutorials/sensor.html> (accessed on 21 March 2022).
27. Dekimpe, R.; Xu, P.; Schramme, M.; Gerard, P.; Flandre, D.; Bol, D. A batteryless ble smart sensor for room occupancy tracking supplied by 2.45 ghz wireless power transfer. *Integration* **2019**, *67*, 8–18. [CrossRef]
28. Gleonec, P.; Ardouin, J.; Gautier, M.; Berder, O. Poster: Multi-source Energy Harvesting for IoT nodes. In Proceedings of the Conference on Design and Architectures for Signal and Image Processing (DASIP), Rennes, France, 12–14 October 2016.
29. Ahn, J. I.; Kim, D. ; Ha, R.; Cha, H. State-of-Charge Estimation of Supercapacitors in Transiently-powered Sensor Nodes. *IEEE Trans. -Comput.-Aided Des. Integr. Circuits Syst.* **2021**, *41*, 225–237. [CrossRef]
30. Gleonec, P.; Ardouin, J.; Gautier, M.; Berder, O. Architecture exploration of multi-source energy harvester for IoT nodes. In Proceedings of the IEEE Online Conference on Green Communications (OnlineGreenComm), Piscataway, NJ, USA, 14 November–17 December 2016; pp. 27–32.
31. Naresh, B.; Singh, V.K.; Sharma, V.K. Flexible Hybrid Energy Harvesting System to Power Wearable Electronics. In Proceedings of the Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, India, 27–28 February 2018; pp. 1–5.
32. Hester, J.; Tobias, N.; Rahmati, A.; Sitanayah, L.; Holcomb, D.; Fu, K.; Bursleson, W. P.; Sorber, J. Persistent Clocks for Batteryless Sensing Devices. *ACM Trans. Embed. Comput. Syst.* **2016**, *15*, 4. [CrossRef]
33. Anagnostou, P.; Gomez, A.; Hager, P.A.; Fatemi, H.; Gyvez, J.P.; Thiele, L.; Benini, L. Torpor: A Power-Aware HW Scheduler for Energy Harvesting IoT SoCs. In Proceedings of the 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS), Platja d'Aro, Spain, 2–4 July 2018; pp. 54–61.
34. Elahi, H.; Munir, K.; Eugeni, M.; Atek, S.; Gaudenzi, P. Energy Harvesting towards Self-Powered IoT Devices. *Energies* **2020**, *13*, 5528. [CrossRef]
35. Lee, H.G.; Chang, N. Powering the IoT: Storage-less and converter-less energy harvesting. In Proceedings of the 20th Asia and South Pacific Design Automation Conference, Chiba, Japan, 19–22 January 2015; pp. 124–129.

36. Zhou, H.; Jiang, T.; Gong, C.; Zhou, Y. Optimal Estimation in Wireless Sensor Networks With Energy Harvesting. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9386–9396. [[CrossRef](#)]
37. Patil, K.; Fiems, D. The value of information in energy harvesting sensor networks. *Oper. Res. Lett.* **2018**, *46*, 362–366. [[CrossRef](#)]
38. Kumar, A.; Saad, W. On the tradeoff between energy harvesting and caching in wireless networks. In Proceedings of the IEEE International Conference on Communication Workshop (ICCW), London, UK, 8–12 June 2015; pp. 1976–1981.
39. Cho, K.; Park, W.; Hong, M.; Park, G.; Cho, W.; Seo, J.; Han, K. Analysis of Latency Performance of Bluetooth Low Energy (BLE) Networks. *Sensors* **2014**, *15*, 59–78. [[CrossRef](#)] [[PubMed](#)]
40. Treurniet, J.J.; Sarkar, C.; Prasad, R.V.; De Boer, W. Energy Consumption and Latency in BLE Devices under Mutual Interference: An Experimental Study. In Proceedings of the 3rd International Conference on Future Internet of Things and Cloud, Rome, Italy, 24–26 August 2015; pp. 333–340.